

AMENDMENT TO THE CLAIMS

This listing of the claims will replace all prior versions, and listings, of claims in the application.

1. (Previously Presented) A gaming method, comprising:
 - receiving a wager from a player via a value input device of a gaming apparatus;
 - generating an image representative of a game for display on a display unit of the gaming apparatus;
 - executing compiled code associated with play of the game using a processor of the apparatus, the compiled code including code to read data from a file stored in a memory of the gaming apparatus, wherein the file is separate from the compiled code;
 - reading, using the compiled code, first uncompiled instruction data from the file, the first uncompiled instruction data indicative of a request to create a first software object associated with play of the game;
 - creating the first software object using the compiled code in response to the first uncompiled instruction data;
 - reading second uncompiled instruction data from the file, the second uncompiled instruction data indicative of a request to create a second software object associated with play of the game;
 - creating the second software object using the compiled code in response to the second uncompiled instruction data;
 - reading third uncompiled instruction data from the file after creation of the second software object, wherein the third uncompiled instruction data includes a reference to the first software object, wherein the third uncompiled instruction data is indicative of a request to perform an operation associated with play of the game and associated with the first software object;

performing, in response to the third uncompiled instruction data, the operation indicated by the third uncompiled instruction data using the compiled code; and determining a value payout associated with an outcome of the game.

2. (Original) A gaming method according to claim 1, further comprising:
saving a name of the first software object in a database of created software objects using the compiled code;
saving a name of the second software object in the database of created software objects using the compiled code.

3. (Original) A gaming method according to claim 2, further comprising:
saving a pointer to the first software object in the database of created software objects using the compiled code;
saving a pointer to the second software object in the database of created software objects using the compiled code.

4. (Original) A gaming method according to claim 1, wherein performing the operation indicated by the third uncompiled instruction data comprises invoking a function of the first software object.

5. (Original) A gaming method according to claim 4, wherein reading the third uncompiled instruction data comprises reading at least one symbol between a name of the first software object and a name of the function that indicates the function is associated with the first software object.

6. (Original) A gaming method according to claim 4, wherein reading the third uncompiled instruction data comprises reading a binary tag indicating that a function belonging to a software object is to be invoked.

7. (Original) A gaming method according to claim 1, wherein performing the operation indicated by the third uncompiled instruction data comprises invoking a function that references the first software object.

8. (Original) A gaming method according to claim 7, wherein reading the third uncompiled instruction data comprises reading at least one symbol indicating the third uncompiled instruction data includes a software object name.

9. (Original) A gaming method according to claim 8, further comprising:
searching a database of created software objects for the name of the first software object using the compiled code; and
retrieving a pointer to the first software object from the database of created software objects if the name of the first software object is included in the database of created software objects.

10. (Original) A gaming method according to claim 8, wherein the at least one symbol indicates the third uncompiled instruction data includes a software object name or a data structure name, wherein the method further comprises:
searching a database of created data structures for the name of the first software object using the compiled code;

searching a database of created software objects for the name of the first software object using the compiled code, if the name is not found in the database of created data structures; and retrieving a pointer to the first software object from the database of created software objects if the name of the first software object is included in the database of created software objects.

11. (Original) A gaming method according to claim 7, wherein reading the third uncompiled instruction data comprises reading a binary tag indicating the third uncompiled instruction data includes a software object name.

12. (Original) A gaming method according to claim 1, further comprising passing an indicator of the first software object to the compiled code after reading the third uncompiled instruction data from the file.

13. (Original) A gaming method according to claim 12, wherein the indicator of the first software object comprises a pointer to the first software object.

14. (Original) A gaming method according to claim 1, wherein the game comprises at least one of a poker game, a blackjack game, a slots game, a keno game, and a bingo game.

15. (Original) A gaming method according to claim 1, wherein at least one of the first uncompiled instruction data, the second uncompiled instruction data, and the third uncompiled instruction data comprise data in a text format.

16. (Original) A gaming method according to claim 1, wherein at least one of the first uncompiled instruction data, the second uncompiled instruction data, and the third uncompiled instruction data comprise data in a binary format.

17. (Original) A gaming apparatus, comprising:

 a display unit;

 a value input device;

 a controller operatively coupled to the display unit and the value input device, the controller comprising a processor and a memory operatively coupled to the processor, wherein the memory includes compiled code and a file separate from the compiled code,

 the controller being programmed to receive wager data from the value input device, the wager data indicative of a wager submitted by a player;

 the controller being programmed to generate an image representative of a game for display on the display unit,

 the controller being programmed according to the compiled code to read first uncompiled instruction data from the file, the first uncompiled instruction data indicative of a request to create a first software object associated with play of the game,

 the controller being programmed according to the compiled code to create the first software object in response to the first uncompiled instruction data,

 the controller being programmed according to the compiled code to read second uncompiled instruction data from the file, the second uncompiled instruction data indicative of a request to create a second software object associated with play of the game,

 the controller being programmed according to the compiled code to create the second software object in response to the second uncompiled instruction data,

the controller being programmed according to the compiled code to read third uncompiled instruction data from the file after creation of the second software object, wherein the third uncompiled instruction data includes a reference to the first software object, wherein the third uncompiled instruction data is indicative of a request to perform an operation associated with play of the game and associated with the first software object,

the controller being programmed according to the compiled code to perform the operation in response to the third instruction data, and the controller being programmed to determine a value payout associated with an outcome of the game.

18. (Original) A gaming method, comprising:

receiving a wager from a player via a value input device of a gaming apparatus;

generating an image representative of a game for display on a display unit of the gaming apparatus;

executing compiled code associated with play of the game using a processor of the apparatus, the compiled code including code to read data from a file stored in a memory of the gaming apparatus, wherein the file is separate from the compiled code;

reading, using the compiled code, first uncompiled instruction data from the file, the first uncompiled instruction data indicative of a request to create a data structure associated with play of the game;

creating the data structure using the compiled code in response to the first uncompiled instruction data;

reading second uncompiled instruction data from the file, wherein the second uncompiled instruction data includes a reference to the data structure, wherein the second uncompiled

instruction data is indicative of a request to perform an operation associated with play of the game and associated with the data structure;

performing, in response to the second uncompiled instruction data, the operation indicated by the second uncompiled instruction data using the compiled code and the data structure; and

determining a value payout associated with an outcome of the game.

19. (Original) A gaming method according to claim 18, wherein reading the first uncompiled instruction data comprises reading at least one symbol indicating the first uncompiled instruction data includes a data structure name corresponding to a data structure to be created.

20. (Original) A gaming method according to claim 18, wherein reading the first uncompiled instruction data comprises reading a binary tag indicating that the first uncompiled instruction data includes a data structure name of a data structure to be created.

21. (Original) A gaming method according to claim 20, wherein the binary tag indicates the second uncompiled instruction includes a data structure name or a software object name.

22. (Original) A gaming method according to claim 18, further comprising saving a name of the data structure in a database of created data structures using the compiled code.

23. (Original) A gaming method according to claim 22, further comprising saving a pointer to the data structure in a database of created data structures using the compiled code.

24. (Original) A gaming method according to claim 18, wherein performing the operation indicated by the second uncompiled instruction data comprises invoking a function, wherein a name of the data structure is a parameter of the function.

25. (Original) A gaming method according to claim 24, wherein reading the second uncompiled instruction data comprises reading at least one symbol indicating that the second uncompiled instruction data includes a data structure name.

26. (Original) A gaming method according to claim 25, further comprising:
searching a database of created data structures for the name of the data structure using the compiled code; and
retrieving a pointer to the data structure from the database of created software objects if the name of the data structure is included in the database of created data structures.

27. (Original) A gaming method according to claim 25, wherein the at least one symbol indicates the second uncompiled instruction data includes a software object name or a data structure name, wherein the method further comprises:

searching a database of created data structures for the name of the data structure using the compiled code;
retrieving a pointer to the data structure from the database of created data structures if the name of the data structure is included in the database of created data structures; and
searching a database of created software objects for the name of the data structure using the compiled code, if the name is not found in the database of created data structures.

28. (Canceled)

29. (Currently Amended) A gaming method according to claim ~~28~~ 25, wherein the at least one symbol indicates the second uncompiled instruction includes a data structure name or a software object name.

30. (Original) A gaming method according to claim 24, wherein reading the second uncompiled instruction data comprises reading a binary tag indicating that the second uncompiled instruction includes a data structure name.

31. (Original) A gaming method according to claim 30, wherein the binary tag indicates the second uncompiled instruction includes a data structure name or a software object name.

32. (Original) A gaming method according to claim 18, further comprising passing an indicator of the data structure to the compiled code after reading the second uncompiled instruction data from the file.

33. (Original) A gaming method according to claim 32, wherein the indicator of the data structure comprises a pointer to the data structure.

34. (Original) A gaming method according to claim 18, wherein the game comprises at least one of a poker game, a blackjack game, a slots game, a keno game, and a bingo game.

35. (Previously Presented) A gaming apparatus, comprising:

a display unit;

 a value input device;

 a controller operatively coupled to the display unit and the value input device, the controller comprising a processor and a memory operatively coupled to the processor, wherein the memory includes compiled code and a file separate from the compiled code,

 the controller being programmed to receive wager data from the value input device, the wager data indicative of a wager submitted by a player;

 the controller being programmed to generate an image representative of a game for display on the display unit,

 the controller being programmed according to the compiled code to read first uncompiled instruction data from the file, the first uncompiled instruction data indicative of a request to create a data structure associated with play of the game,

 the controller being programmed according to the compiled code to create the data structure in response to the first uncompiled instruction data,

 the controller being programmed according to the compiled code to read second uncompiled instruction data from the file, wherein the second uncompiled instruction data includes a reference to the data structure, wherein the second uncompiled instruction data is indicative of a request to perform an operation associated with play of the game and associated with the data structure,

 the controller being programmed according to the compiled code to perform the operation in response to the second instruction data using the data structure, and

 the controller being programmed to determine a value payout associated with an outcome of the game.